

Load. Наконец, мы загружаем обработанные данные в базу MySQL. Полученный набор данных используется для генерации отчетов (пример отчета на Рис. 3), которые затем отправляются рекламодателям.

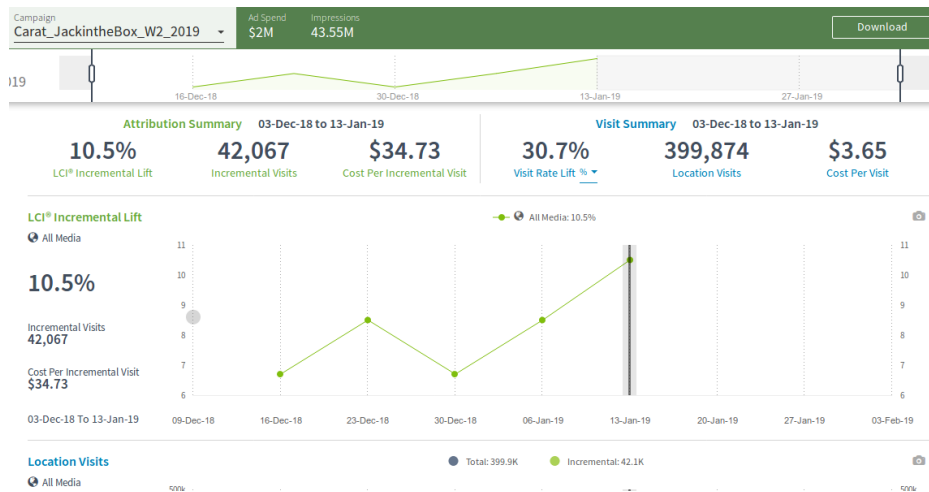


Рис 3. Пример отчета о рекламной кампании

Вывод

Данная система обрабатывает более полумиллиона событий в минуту, обрабатывает их и выдает отчет в форме, удобной для клиентов. В результате клиент может оценить эффективность рекламной кампании. Планируется перевод ключевых модулей с языка Python на Scala, что позволит ускорить работу системы и увеличить отказоустойчивость.

Список использованных источников:

1. Breur, Tom. "Statistical Power Analysis and the contemporary "crisis" in social sciences". *Journal of Marketing Analytics*, July 2016 – 65 с.
2. Loshin, D. *Business Intelligence* - 2nd / Loshin, D. - Morgan Kaufmann, 2012. - 400 с.
3. Stange, M. *Real-Time Advertising. Business & Information Systems Engineering* / M. Stange, B. Funk. - Fakultät Wirtschaftswissenschaften Leuphana University Lüneburg Lüneburg Germany, 2014. - 308 с.
4. White, T. *Hadoop: The Definitive Guide, Fourth Edition* / T. White. – Gravenstein Highway North, Sebastopol: O'Reilly Media, 2015. – 728 с.
5. Karau, H. *Learning Spark* / h. Karau, A. Konwinski, P. Wendell, M. Zaharia. – Gravenstein Highway North, Sebastopol: O'Reilly Media, 2015. – 255 с.
6. GitHub [Электронный ресурс] / Luigi. - Режим доступа : <https://github.com/spotify/luigi>. - Дата доступа : 23.12.2018.

ГИБРИДНАЯ СТРАТЕГИЯ ОТОБРАЖЕНИЯ ЛОГИЧЕСКИХ АДРЕСОВ В ФИЗИЧЕСКИЕ В НАКОПИТЕЛЯХ ИНФОРМАЦИИ НА ОСНОВЕ ФЛЭШ-ПАМЯТИ

Шиманский И.Д.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Заливако С.С. – кандидат техн. наук

Статья посвящена проблеме отображения логических адресов в физические в накопителях информации на основе NAND-технологии флэш-памяти. Выделяются и описываются характерные особенности основных стратегий отображения. Особое внимание уделено гибридной стратегии отображения. Дается сравнение всех рассмотренных стратегий. В качестве ключевого свойства для сравнения используется объем динамической памяти, требуемый для хранения таблиц отображения. Представлены достоинства и недостатки гибридной стратегии относительно остальных, а также вектор работы для более детального изучения её характеристик.

В данной работе в качестве примера устройства используется Samsung K9XXG08UXM [1]. Необходимые характеристики приведены в таблице 1. Размер логической страницы, используемой файловой системой, будем считать равной 4КБ. Стоит подчеркнуть, что в данном примере предполагается, что размеры физических и логических страниц равны.

Таблица 1 – Характеристики Samsung K9XXG08UХМ.

Характеристика	Значение	Характеристика	Значение
Объем данных страницы	4КБ	Количество страниц в блоке	64
Объем метаданных страницы	128Б	Количество блоков в plane	2 048
Объем данных блока	256КБ	Количество plane в устройстве	16
Общий объем диска	8ГБ	Общее количество страниц	2 097 152

К задаче отображения адресов можно также отнести сохранение необходимых данных в энергонезависимую память для возможности последующего восстановления всей структуры отображения в случае внезапного отключения питания. Для этих целей можно использовать как отдельные блоки, так и зарезервированное пространство в каждой физической странице. Для простоты будет использоваться второй способ.

Для сравнения рассмотрим сначала два простых метода отображения адресов: отображение на уровне страниц и отображение на уровне блоков [2].

В первом случае каждому номеру логической страницы (*lpn*) ставится в соответствие номер физической страницы (*ppn*). Размер таблицы пропорционален общему количеству страниц в устройстве. Запись данных производится последовательно во все страницы активного блока с соответствующими изменениями в таблице отображения для каждой записанной страницы.

В случае отображения на уровне блоков, логические адреса разбиваются на равные блоки (*lbn*), каждому из которых соответствует номер физического блока (*pbn*). Смещение страницы в логическом и физическом блоках равны. Определить номер блока и смещение в нём можно как показано в соотношении (1):

$$lbn = [lpn / n], \quad (1)$$

$$offset = lpn \bmod n,$$

где *n* – количество страниц в блоке.

Таблица пропорциональна количеству блоков в устройстве. При изменении даже одной страницы необходимо перезаписывать весь блок, что быстро изнашивает элементы памяти.

Основная идея гибридной стратегии заключается в комбинировании таблиц блочного и страничного уровней. Большая часть физических блоков представлена на блочном уровне (*data-blocks*), что обеспечивает малый требуемый объём памяти по сравнению с отображением на уровне страниц. Для обеспечения эффективных записей размера меньше блока, дополнительно выделяется пул блоков (*log-blocks*), для которых ведётся таблица со страничной гранулярностью.

На рисунке 1 показан пример последовательности из семи записей. Предполагается наличие только четырёх страниц в блоке. Во время первой операции записи логическому блоку 0 назначается физический блок в таблице *data-blocks*. Производится запись данных по необходимому смещению в блоке, и в метаданных первой страницы блока записывается номер соответствующего логического блока. Вторая запись также производится непосредственно в *data-block*, так как страница с нужным смещением свободна. Третья запись производится уже в *log-block*, так как эта страница уже была записана ранее. Для этого логическому блоку 0 дополнительно назначается *log-block* со страничной гранулярностью из пула. Записи в *log-block* производятся до полного его заполнения. В метаданных страницы *log-block* записывается смещение страницы в логическом блоке, которую он представляет. Записи 4-7 производятся аналогично. После того как *log-block* полностью записан, происходит объединение его с соответствующим блоком данных. Результат объединения записывается в чистый *data-block*. После чего использованные *data-block* и *log-block* могут быть очищены и использоваться повторно.

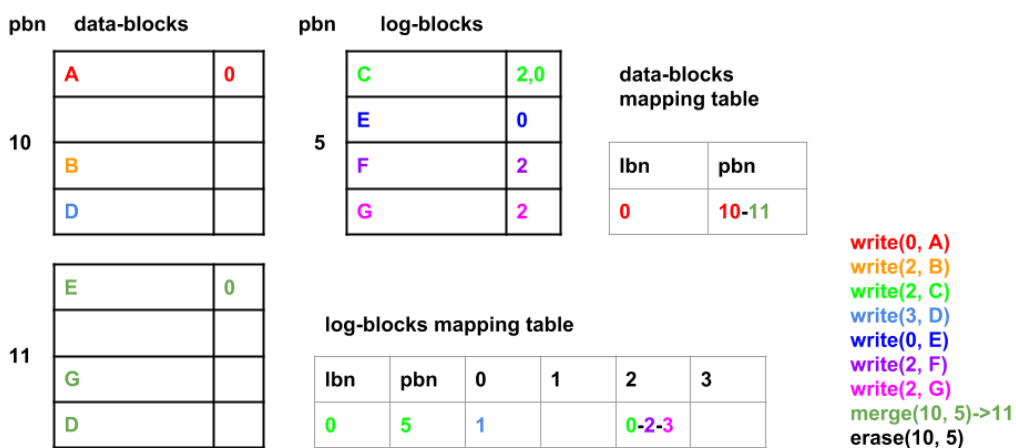


Рисунок 1 – Гибридная стратегия отображения.

Количество блоков в пуле должно быть ограничено, мы будем отталкиваться от избыточного объема памяти накопителя (overprovisioning). В среднем overprovisioning составляет 10% от общего объема памяти [3], половину от него выделим для поддержания пула (5%), что приближенно составляет 100 блоков на plane (подмножество блоков, физически размещенных на одной интегральной схеме), или 1 600 на всё устройство. Имеет смысл равномерно распределять блоки из пула по plane, так как большинство контроллеров NAND поддерживают операцию обратного копирования (copy-back), которая позволяет копировать данные из одной страницы в другую на том же plane [1]. Эта операция читает данные во внутренний регистр данных и записывает из него же, обеспечивая лучшую производительность за счёт отсутствия передачи данных по шине. Если в пуле не остаётся чистых блоков, выбирается наиболее заполненный блок из пула, после чего объединяется с соответствующим блоком данных и очищается.

Необходимое количество памяти рассчитывается исходя из размера одного элемента таблицы и количества записей. Для страничной стратегии достаточно 3Б памяти на элемент таблицы, чтобы хранить номер физической страницы. Количество записей таблицы равно количеству страниц. Для блочной стратегии – 2Б, чтобы хранить номер физического блока. Количество записей равно количеству блоков. Для гибридной стратегии одна таблица идентична блочной. Каждая запись второй таблицы хранит номер логического и физического блоков (2Б + 2Б) и 64 однобайтовых смещения страниц в физическом блоке (64Б). Количество записей второй таблицы, как было показано ранее, равно 1 600. Общий объём памяти двух таблиц: 64КБ + 106КБ = 170КБ.

Таблица 2 – Потребление памяти разных стратегий.

	Страничная	Блочная	Гибридная
Размер элемента таблицы	3Б	2Б	2Б, 68Б
Количество записей	2 097 152	32 768	32 768, 1 600 (~5%)
Размер таблицы	6МБ	64КБ	170КБ

Из таблицы 2 видно, как и ожидалось, меньше всего памяти потребляет блочная стратегия. Также можно отметить, что гибридная стратегия обеспечивает более эффективные записи размера страницы относительно блочной стратегии при гораздо меньшем потреблении памяти относительно страничной стратегии.

К недостаткам можно отнести большую задержку доступа к памяти, так как надо произвести поиск в двух таблицах в случае если log-block и data-block ещё не были объединены, и потребность в дополнительной памяти накопителя для поддержания пула блоков.

Для более детальной оценки рассмотренных стратегий в дальнейшем планируется реализовать их в системе моделирования флэш-памяти “MQSim”. Иначе сложно оценить такие факторы, как задержка выполнения разных операций, сборка мусора, выравнивание износа. Даже те объёмы памяти, которые были рассчитаны в статье, в большинстве случаев не могут храниться в динамической памяти полностью, поэтому необходимо решать задачу кэширования таблиц.

Список использованных источников:

1. Design Tradeoffs for SSD Performance / Nitin Agrawal [et al] // USENIX Technical Conference – June 2008.
2. Reconfigurable FTL (Flash Translation Layer) Architecture for NAND Flash-Based Application / Chanik Park [et al] // ACM Transactions on Embedded Computing Systems – July 2008 – Vol. 7 – No. 4 – Article 38.
3. Error Characterization, Mitigation, and Recovery in Flash-Memory-Based Solid-State Drives / Yu Cai [et al] // IEEE – September 2017.

ПРИЛОЖЕНИЕ ДЛЯ УВЕЛИЧЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ ТРУДА НА ОСНОВЕ ТЕХНИКИ POMODORO

Чачура Р.В., Соболевский С.Е.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Жвакина А.В. – к.т.н., доцент

Во время разработки произведены исследования в области информационных технологий, в частности инструментов для разработки серверной и пользовательской части приложения по улучшению умственной производительности человека.

Increaser – это приложение для тех, кому для выполнения работы необходима умственная концентрация: студенты, инженеры, программисты, математики и другие технические работники. Эти люди хотят делать больше глубокой, целенаправленной работы и меньше мелкой, полной